Fordham Law Review

1991

Copyright Protection of SSO: Replete with Internal Deficiencies and Practical Dangers

Cary S. Kappel

Copyright Protection of SSO: Replete with Internal Deficiencies and Practical Dangers

Cover Page Footnote

I would like to thank Professor Hugh Hansen for his help in writing this Note, particularly with respect to the analysis of the Whelan and Lotus decisions.

COPYRIGHT PROTECTION OF SSO: REPLETE WITH INTERNAL DEFICIENCIES AND PRACTICAL DANGERS*

INTRODUCTION

Since the enactment of the Copyright Act of 1976,[1] the literal elements of computer programs[2] have clearly been subject to copyright protection.[3] Over the last five years, however, courts have shown an increasing willingness to expand the protection of computer programs beyond their literal elements. In some jurisdictions, protection has been extended to the "sequence, structure, and organization" ("SSO")[4] and even to the "display screens"[5] of computer software. At least one commentator has urged that protection be extended to custom and semi-custom logic devices.[6]

---

1. Pub. L. 94-553, 90 Stat. 2541 (codified as amended at 17 U.S.C. §§ 101-810 (1988)).

2. The literal elements of a computer program are the actual instructions contained in the program.

3. *See* H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S. Code Cong. & Admin. News 5659, 5667.

4. *See* Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1240 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987); Johnson Controls, Inc. v. Phoenix Control Sys., Inc., 886 F.2d 1173, 1175-76 (9th Cir. 1989). The sequence, structure and organization of a program is "the manner in which the program operates, controls and regulates the computer in receiving, assembling, calculating, retaining, correlating, and producing useful information." *Whelan*, 797 F.2d at 1239 (quoting Whelan Assocs., Inc., v. Jaslow Dental Laboratory, Inc., 609 F. Supp. 1307, 1320 (E.D. Pa.), *amended in part*, 609 F. Supp. 1325 (E.D. Pa. 1985), *aff'd*, 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987)).

5. *See, e.g.*, Manufacturers Technologies, Inc. v. Cams, Inc., 706 F. Supp. 984, 996 (D. Conn. 1989)("plaintiff's screen displays . . . merit protection as copyrightable expression"); Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1133 (N.D. Cal. 1986)(rejecting "defendant's argument that the overall structure, sequencing, and arrangement of screens in [the program] fall outside the ambit of copyright protection"). *But see* Digital Communications Assocs., Inc. v. Softklone Distrib. Corp., 659 F. Supp. 449, 455, 463, 465 (N.D. Ga. 1987)(rejecting concept that a *program* can be infringed through copying of its display screens, but holding that display screens are separately copyrightable as a literary work/compilation). Thus, protection of "display screens" goes beyond SSO protection because it is protection completely divorced from the literal elements of the program and the program's structure. *See generally* Note, *Copyright Protection for Computer Screen Displays*, 72 Minn. L. Rev. 1123-24 (1988)(discussing copyright protection of display screens). It should also be noted that the term "display screen" is often used interchangeably with "user interface."

6. *See* Lunney, *Copyright Protection for ASIC Gate Configurations: PLDs, Custom and Semicustom Chips*, 42 Stan. L. Rev 163, 163 (1989). ASICs (Application Specific Integrated Circuits) are defined by the author as including PLDs (Programmable Logic Devices), custom and semi-custom chips. *See id.* at 165. PLDs "consist of a paired array of AND and OR gates. . . . Custom and semicustom chips have gates configured to achieve the logic pattern desired by a purchaser." *Id.* at 168. For a more complete discussion of ASICs, see generally S. Shiva, Introduction to Logic Design 292-330 (1988)(discussing PLDs, custom logic devices, and other forms of programmable logic).

This Note argues that the trend towards expanded protection is deficient both in its interpretation of copyright law and in its practical implications. Part I of this Note briefly discusses the general principles of copyright law. Part II of this Note discusses existing case law on the copyrightability of computer programs, both in general and with respect to the protection of SSO. Part III examines the statutory and practical bars to SSO protection. Part IV will demonstrate that literal copyright protection combined with patent protection is adequate to serve the goals of intellectual property law.[7]

I. THE LAW OF COPYRIGHT

The power to create copyrights and patents was granted to Congress in Art. I, § 8, cl. 8 of the Constitution.[8] Congress has exercised this power through the enactment of various copyright statutes, culminating with the Copyright Act of 1976 ("Act").[9]

Under Section 102 of the Act, protection "subsists . . . in original works of authorship fixed in any tangible medium of expression ."[10] This section additionally defines works of authorship to include literary works; musical works, including any accompanying words; dramatic works, including any accompanying music; pantomimes and choreographic works; pictorial, graphic, and sculptural works; motion pictures and other audiovisual works; and sound recordings.[11] The Act further provides that computer programs are to be protected under Section 102 as "literary works."[12] In 1980, in response to the National Commission on New Technological Uses of Copyrighted Works Report,[13] the Act

---

7. Another alternative is to create separate legislation, outside of both copyright and patent law, to deal with this problem. This course, which may be the most practical, is nevertheless beyond the scope of this Note. *See generally* Raskind, *Symposium: The Future of Software Protection: The Uncertain Case for Special Legislation Protecting Computer Software*, 47 U. Pitt. L. Rev. 1131 (1986)(proposing specific legislation for protection of computer software).

8. "The Congress shall have Power . . . To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors, the exclusive Right to their respective Writings and Discoveries." U.S. Const. art. I, § 8, cl. 8.

9. Pub. L. 94-553, 90 Stat. 2541 (codified as amended at 17 U.S.C. §§ 101-810 (1988)).

10. 17 U.S.C. § 102 (1988).

11. *See id.*

12. *Id.* § 102a. *See also* H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S. Code & Admin. News 5659, 5667 ("The term 'literary works' . . . includes . . . computer programs.").

13. In 1978, after three years of investigation, the National Commission on New Technological Uses of Copyrighted Works submitted its final report. *See* Final Report of the National Commission on New Technological Uses of Copyrighted Works (1979) [hereinafter "CONTU"]. The Commission was formed to review the law of copyright concerning computer programs and to make recommendations on new legislation. *See id.* at 1. The report is generally considered the legislative history of the 1976 Copyright Act with regard to computer programs. *See, e.g.,* Vault Corp. v. Quaid Software Ltd., 847 F.2d 255, 261 (5th Cir. 1988)("[C]ourts . . . rely on the CONTU Report as an expression of legislative intent."); Micro-Sparc, Inc. v. Amtype Corp., 592 F. Supp. 33, 35 n.7

was amended to further define a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."[14]

Although the Act grants copyright protection to computer programs, Section 102(b) of the Act limits the protection afforded to all works of authorship by providing that "[i]n no case does copyright protection . . . extend to any idea, procedure, process, system [or] method of operation."[15]

To determine whether the non-literal elements of computer programs fall within the subject matter of copyright as defined by Section 102 of the Act, three basic doctrines must be examined: the process/expression dichotomy, the idea/expression dichotomy and the merger doctrine.

The process/expression dichotomy[16] traces its roots to *Baker v. Selden*.[17] In that case, the plaintiff had written a book explaining a new

---

(D. Mass. 1984)(CONTU has been considered an expression of legislative intent); Midway Mfg. Co. v. Strohon, 564 F. Supp. 741, 750 n.6 (N.D. Ill. 1983)(same); Note, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 Mich. L. Rev. 866, 886 (1990)("[S]everal courts have considered the CONTU Report a reflection of legislative intent—even with respect to the 1976 Act."); Comment, *The Incompatibility of Copyright and Computer Software: An Economic Evaluation and a Proposal for a Marketplace Solution*, 66 N.C.L. Rev. 977, 984 (1988)(courts look to CONTU for legislative history). *But see* Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1242 (3d Cir. 1986)(CONTU is legislative history only as to those provisions amended in response to the Report), *cert. denied*, 479 U.S. 1031 (1987).

Unfortunately, the report does not explicitly address the protection of non-literal elements of computer programs. It does, however, discuss the difficulty in distinguishing between process and expression. *See* CONTU, *supra*, at 22; Note, *supra*, at 888; *infra* note 16 and accompanying text.

Due to the absence in this discussion of anything remotely resembling the protection of SSO, it could be argued that the Commission did not envision such protection. One author noted that "[o]ne view is that CONTU did not contemplate that the copying of anything other than literal instructions could ever constitute copyright infringement." Note, *supra*, at 888. The author went on to assert that "[a]n alternative interpretation is that CONTU never expressed a view on whether program structure should be protected." *Id.* at 889.

Regardless of whether CONTU implicitly rejected SSO protection or was merely ambiguous as to its protection, there are far more important reasons to reject non-literal protection. *See infra* notes 50-85 and accompanying text.

14. Pub. L. 96-517, § 10(a), 94 Stat. 3028 (codified as amended 17 U.S.C. § 101 (1988)). In addition, Congress amended Section 117 of the Act in response to the CONTU Report. *See id.* (codified as amended at 17 U.S.C. § 117 (1988)). A discussion of Section 117, however, is beyond the scope of this Note.

15. Pub. L. 94-553, § 102(b), 90 Stat. 2544 (codified as amended at 17 U.S.C. § 102(b)). It should be noted that this preclusion existed at common law prior to the Act's passage. *See* Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 741-42 (9th Cir. 1971); Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931). This preclusion is discussed more fully later in this Note. *See infra* notes 23-25 and accompanying text.

16. *See* Note, *supra* note 13, at 876; *see also* CONTU, *supra* note 13, at 22 (discussion of the process/expression distinction).

17. 101 U.S. 99 (1879).

system of accounting that he had devised.[18] Within the book were sample forms with which to use this method of accounting.[19] The defendant produced a similar form that was designed to be used with this system of accounting.[20] The Court held that although the plaintiff's system of accounting could not be protected, his book describing the system could be protected.[21] Therefore, an expression of a process is copyrightable, but the process itself is not.[22]

Similarly, the idea/expression dichotomy posits that only the expression of an idea can be protected, not the idea itself.[23] It is thus necessary to determine how much of the material sought to be protected is idea and how much is expression.[24] The dichotomy has been developed mostly with regard to the plots of literary works, where the plot of the infringing work is sufficiently similar to the copyrighted work to constitute a copying of expression. The judgment is generally made on a case-by-case basis using Judge Learned Hand's levels-of-abstraction test.[25]

The merger doctrine holds that where an idea can be expressed in only one way, there can be no protection of the expression because to protect the expression would be to protect the idea.[26] These three doctrines provide the guidelines for analyzing SSO protection.

II. THE CASE LAW

A.   *The Early Cases—Protection of the Literal Elements*

The early cases involving computer programs under the Act dealt only with literal copying of computer code. In *Apple Computer, Inc. v. Frank-*

---

18. *See id.* at 99-100.

19. *See id.*

20. *See id.* at 100.

21. *See id.* at 103-104.

22. *See id.* at 104-05.

23. *See* Reyher v. Children's Television Workshop, 533 F.2d 87, 90-91 (2d Cir.), *cert. denied*, 429 U.S. 980 (1976); Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 741-42 (9th Cir. 1971); Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931).

24. Because computer programs are defined as literary works under the Act, many courts have attempted to define the "expression" of computer programs by adopting tests developed for determining the expression of novels. *See, e.g.,* Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1233-34 (3d Cir. 1986)(making an analogy between non-literal elements of books and the non-literal elements of computer programs), *cert. denied*, 479 U.S. 1031 (1987); Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 54 (D. Mass. 1990)(same).

25. *See infra* note 38.

26. *See* Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 741-42 (9th Cir. 1971); *see, e.g.*, Morrissey v. Procter & Gamble Co., 379 F.2d 675, 678-79 (1st Cir. 1967)("When the uncopyrightable subject matter is very narrow, so that 'the topic necessarily requires,' . . . at best only a limited number" of forms of expression, there can be no copyrighting of the expression)(citations omitted); Crume v. Pacific Mut. Life Ins. Co., 140 F.2d 182, 184 (7th Cir.), *cert. denied*, 322 U.S. 755 (1944)(there can be no copyright protection where "the practical use of the art explained by the copyright and lodged in the public domain can be attained solely by the employment of language which gives expression to that which is disclosed").

lin Computer Corp.,[27] the court held computer programs copyrightable regardless of whether they were in source or object code[28] and regardless of whether the programs were application programs or operating system programs.[29] The court seemed to foreclose the possibility of SSO protection by stating in dicta that Apple would not be able to "copyright the method which instructs the computer to perform its operating functions."[30]

Similarly, in *Synercom Technology, Inc. v. University Computing Co.*,[31] a district court held that "the sequence and ordering" of input data was idea rather than expression and therefore unprotectable.[32] This holding was later relied upon by the Fifth Circuit to deny protection to SSO.[33]

Copyright law also extends protection against literal translations from one computer language to another. In *E.F. Johnson Co. v. Uniden Corp. of Am.*,[34] the court concluded that although the infringing program was not a line-by-line copy of the original, the fact that the infringing work was essentially a translation warranted a finding of infringement even though the plaintiff and defendant used different microprocessors and thus different assembly languages.[35] None of these cases, however, indicated that protection could extend beyond the literal elements.

B.   *The Expansion into Protection of Sequence Structure And Organization*

In 1986, copyright protection was extended beyond the literal code to the "sequence, structure and organization" (SSO) of computer programs

---

27. 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

28. *See id.* at 1249. The source code is the program as written by the programmer in human readable form. The object code is the original program translated into binary code (e.g. 10001101) that the machine can understand. *See* Note, *supra* note 13, at 869 n.15.

29. *See Apple Computer, Inc.*, 714 F.2d at 1252. An application program is what most people associate with a computer program. It is a program "run" on a computer that performs a specific task for the user, such as spread sheet programs and computer games. An operating system program is what controls the internal functioning of the computer as a whole. It is essentially a part of the hardware itself and is sometimes embedded in the hardware in the form of a ROM (Read Only Memory). *See* Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stan. L. Rev. 1045, 1048 (1989).

30. *Apple Computer, Inc.*, 714 F.2d at 1251.

31. 462 F. Supp. 1003 (N.D. Tex. 1978).

32. *See id.* at 1013-14.

33. *See infra* notes 42-44 and accompanying text.

34. 623 F. Supp. 1485 (D. Minn. 1985).

35. *See id.* at 1497. This decision, which dealt with a program embedded in a mobile radio unit, is also interesting because it extended protection to a program that was integrally related to its hardware environment. *See infra* note 75 and accompanying text. Another example of protection of this type of program is found in *NEC Corp. v. Intel Corp.*, 645 F. Supp. 590 (N.D. Cal. 1986), in which the court extended protection to the "microcode" of an Intel 8086 microprocessor. The microcode is a program that directs the control section of a microprocessor to interpret or execute instructions. *See id.* at 592-93.

by the Court of Appeals for the Third Circuit in *Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc.*[36] The court reasoned that because copyright law protects non-literal elements in other literary works, such as the plots of novels, through the idea/expression dichotomy, it should also protect non-literal elements in computer programs.[37] *Whelan* interpreted the idea/expression dichotomy to mean that "the purpose or function of . . . [the] work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea"[38] and therefore protected.

Subsequently, in *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*,[39] the Court of Appeals for the Ninth Circuit, without citing *Whelan*, held that "[w]hether the non-literal components of a program, including the structure, sequence and organization and user interface, are protected depends on whether, on the particular facts of each case, the component in question qualifies as an expression of an idea, or an idea itself."[40] The court did not, however, adopt the broad standard of *Whelan*, which declared that only the basic purpose of the program is unprotected. It did not explicitly adopt any test for distinguishing idea from expression. It noted, however, that several unique features of the plaintiff's program supported the proposition that the structure was "expression."[41]

---

36. 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987). In this case the plaintiff, Whelan, was an experienced developer of computer software and the defendant, Jaslow, manufactured dental prosthetics and devices. Jaslow sought to have his business operations computerized and hired Whelan to write the software. Shortly after Whelan developed the program, he decided to market it publicly, and Jaslow became the sales representative for the Dentalab Program. Jaslow later wrote a program called Dentcom, which was written in a different computer language than Dentalab and was advertized as a new version of Dentalab. Jaslow sued for misappropriation of trade secrets and Whelan counterclaimed for copyright infringement. *See id.* at 1225-27.

37. This doctrine holds that only the expression of the idea is protected, not the idea itself. *See supra* notes 23-25 and accompanying text.

38. *Whelan*, 797 F.2d at 1236 (emphasis omitted). The court held that the "idea" of the Dentalab program was "simply to run a dental laboratory in an efficient way." *Id.* at 1238 n.34. Therefore, it concluded that the SSO, which was not necessary to this idea, was protected. The *Whelan* court's definition of the idea/expression dichotomy should be contrasted to the more restrictive, albeit less precise, test of Judge Learned Hand that has served as one of the major tools for distinguishing idea from expression:

> Upon any work, and especially upon a play, a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the play is about . . . but there is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his "ideas."

Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902, 902-03 (1931) [hereinafter "Learned Hand Test"].

39. 886 F.2d 1173 (9th Cir. 1989).

40. *Id.* at 1175. It should be noted that the court included under the scope of protection the "user interface." The copyrightability of user interfaces is another hotly disputed issue in copyright law. *See supra* note 5 and accompanying text.

41. 886 F.2d at 1176.

The extension of copyright protection to the non-literal elements of computer programs was rejected by the Court of Appeals for the Fifth Circuit in *Plains Cotton Cooperative Ass'n v. Goodpasture Computer Serv., Inc.*,[42] which, citing a district court case[43] that held input formats to be idea rather than expression, refused to extend protection to SSO. After noting that the district court case was contrary to *Whelan*, the *Plains Cotton* court stated simply: "We decline to embrace *Whelan*."[44]

Despite the Fifth Circuit's rejection of *Whelan*, the general trend seems to be towards adoption of SSO protection in some form.[45] In *Broderbund Software, Inc. v. Unison World, Inc.*,[46] the District Court for the Northern District of California declared that "this [c]ourt is persuaded by the reasoning of *Whelan*," which "stands for the proposition that copyright protection is not limited to the literal aspects of a computer program, but rather that it extends to the overall structure of a program."[47] In *Lotus Dev. Corp. v. Paperback Software Int'l*,[48] the District Court of Massachusetts, in a long and ultimately confusing opinion,

---

42. 807 F.2d 1256 (5th Cir.), *cert. denied*, 484 U.S. 821 (1987).

43. *Id.* at 1262 (citing Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003 (N.D. Tex. 1978)).

44. *Id.*

45. *See, e.g.,* Bull HN Information Sys., Inc. v. American Express Bank Ltd., Copyright L. Dec. (CCH) ¶26,555, at 23,279 (S.D.N.Y. 1990)(endorsing *Whelan's* protection of SSO); Pearl Sys., Inc. v. Competition Elecs., Inc., 8 U.S.P.Q.2d (BNA) 1520, 1524, Copyright L. Dec. (CCH) ¶26,338, at 22,174 (S.D. Fla. 1988)("Pearl Systems device captures the 'total concept and feel' of the Competition Electronics subroutines. Copyright protection of computer software is not limited to the text of the source code or the object code.")(citations omitted); Healthcare Affiliated Servs., Inc. v. Lippany, 701 F. Supp. 1142, 1150-52 (W.D. Pa. 1988)(endorsing *Whelan* but denying protection because no evidence of SSO was presented at trial); Dynamic Solutions, Inc. v. Planning & Control, Inc., Copyright L. Dec. (CCH) ¶26,062, at 20,912 (S.D.N.Y. 1987)("I agree with the Third Circuit's careful analysis in *Whelan Associates*").

46. 648 F. Supp. 1127 (N.D. Cal. 1986).

47. *Id.* at 1133. In fact, the court in *Broderbund* read *Whelan* as extending protection not only to SSO, but also to the "display screens" of the program. *See id.* This has caused concern even for those in favor of SSO protection. One author noted:

> the *Broderbund* court has paved the way for a new and unanticipated application of *Whelan*. Under *Broderbund*, software designers will not be able to market programs which use the same, or a substantially similar, user interface . . . .
> [T]he . . . decision extended copyright protection to a program's menu screens *without regard for the lack of similarity in the underlying code.*

Casenote, Broderbund Software, Inc. v. Unison World, Inc.: *Confusing the "Look" and the "Feel" of Computer Software in Copyright Infringement Cases,* 2 Software L.J. 113, 121 (1987)(emphasis added); *see also* Comment, *supra* note 13, at 992-94(criticizing the expansion of copyright protection to display screens). *But see* Note, *supra* note 5, at 1124 (endorsing copyright protection of screen displays); Note, *A Thousand Clones: The Scope of Copyright Protection in the "Look and Feel" of Computer Programs—*Digital Communications Assocs., Inc. v. Softklone Distributing Corp., 659 F. Supp. 449 (N.D. Ga. 1987), 63 Wash. L. Rev. 195, 220 (1988)(same). A detailed discussion of the "display screen" cases is beyond the scope of this Note. The *Broderbund* decision, however, demonstrates how far things have progressed since *Whelan* first opened the door to non-literal program protection.

48. 740 F. Supp. 37 (D. Mass. 1990).

FORDHAM LAW REVIEW [Vol. 59

also endorsed the protection of SSO.[49]

---

49. In a fifty-page decision, the District Court of Massachusetts held that the "menu command structure" of the Lotus 1-2-3 spreadsheet program had been infringed by the defendant's VP Planner program. *See id.* at 68-69. The decision is as flawed as it is long. The court, apparently unfamiliar with copyright doctrine, devoted nearly six pages to a discussion of useful articles. *See id.* at 52-58. Computer programs, however, are indisputably defined as "literary works" under the Act, see *supra* note 12 and accompanying text, and thus the law on useful articles is completely irrelevant, because it applies only to "pictorial, graphic, and sculptural works." For a discussion of useful articles, see the definitions of "useful articles" and "pictorial, graphic and sculptural works," 17 U.S.C. § 101 (1988).

Even more distressing, however, was the court's insistence that copyright law imposes a condition of novelty or non-obviousness. *See Lotus,* 740 F. Supp. at 58-59 (It should be noted that the court uses the term non-obvious in its lay sense, which is synonymous with the ordinary meaning of "novelty," and does not attach to it the more specific patent definition of non-obviousness. *See id.; infra* note 67). The court was apparently misled by an errant decision of the District Court for the Eastern District of Pennsylvania. *See Lotus,* 740 F. Supp. at 59 (citing E.H. Tate Co. v. Jiffy Enters., Inc., 16 F.R.D. 571, 573 (E.D. Pa. 1954)). No such condition of non-obviousness or novelty exists in copyright. *See, e.g.,* Apple Barrel Prods., Inc. v. R.D. Beard, 730 F.2d 384, 388 n.8 (5th Cir. 1984)("[n]ovelty is not required in a work in order for the work to be . . . copyrightable"); Alfred Bell & Co. v. Catalda Fine Arts, Inc., 191 F.2d 99, 102-03 (2d Cir. 1951)(quoting Baker v. Selden, 101 U.S. 99, 102-03 (1879))("The novelty of the art or thing described or explained has nothing to do with the validity of the copyright."). The court's intermingling of the concepts of non-obviousness and expression continue throughout the opinion, ultimately rendering its conclusions doctrinally deficient. *See Lotus,* 740 F. Supp. at 65-68.

To its credit, the court, unlike the Third Circuit in *Whelan,* did attempt to formulate a test to differentiate idea from expression. Unfortunately, the court's apparent lack of expertise taints this aspect of the decision as well. The test is formulated as follows:

> FIRST, . . . the court [must] conceive, along the scale from the most generalized conception to the most particularized, . . . some conception or definition of the "idea"—for the purpose of distinguishing between the idea and its expression.

Id. at 60 (emphasis omitted). If the idea/expression dichotomy applies to computer programs, this is a logical first prong for determining its application—it is, in effect, a restatement of the Learned Hand Test. *See* Learned Hand Test, *supra* note 38.

> SECOND, the decisionmaker must focus upon whether an alleged expression of the idea is limited to elements essential to expression of that idea (or is one of only a few ways of expressing the idea) or instead includes identifiable elements of expression not essential to every expression of the idea.

Lotus, 740 F. Supp. at 61 (emphasis omitted). This prong, again presupposing the applicability of idea/expression, is a valid rewording of the merger doctrine, which merely states that where a certain element of the idea can be expressed in only one way, that expression cannot be protected.

> THIRD, having identified elements of expression not essential to every expression of the idea, the decisionmaker must focus on whether those elements are a substantial part of the allegedly copyrightable "work."

Id. In this third prong, the court's inexperience with copyright doctrine re-emerges. When determining *copyrightability,* the amount of the overall work that constitutes expression is irrelevant because all expression is copyrightable. *See* Alfred Bell & Co. Ltd. v. Catalda Fine Arts, Inc., 191 F.2d 99, 102-03 (2d Cir. 1951)("All that is needed . . . is that the 'author' contributed . . . something recognizably 'his own' "). The court is confusing the requirement of an "unlawful appropriation" in order to succeed in an infringement action with the requirements of copyrightability. *See infra* note 87 and accompanying text.

Oddly, having expended much time and effort into developing this test, the court applied it in a very vague fashion. *See Lotus,* 740 F. Supp. at 63-70. The court finally

C. *The Internal Deficiencies Of The* Whelan *Doctrine*

Although *Plains Cotton* wisely refused to extend protection to the SSO of computer programs, it failed to properly ground its decision in copyright doctrine. In fact, neither *Plains Cotton* nor *Johnson Controls* expended much effort in evaluating copyright doctrine. In contrast, *Whelan* expended a considerable amount of effort, but its conclusions have serious shortcomings.

In *Whelan*, the court attempted to justify SSO protection through the use of the idea/expression dichotomy,[50] analogizing the SSO to the plot of a book.[51] This approach is problematic, however, because the idea/expression dichotomy was created to deal with literary works such as novels and is ill-suited for use in the area of computer programs. The SSO of a computer program is conceptually closer to a system of accounting, which was held not copyrightable in *Baker v. Selden*,[52] than to the plot of a novel. Even if the analogy proposed by *Whelan* were accepted, however, the court's application of the idea/expression dichotomy is highly suspect.

The court proposed that "the purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea."[53] The court derived this test from the following passage in *Baker v. Selden*:

> "where the art [*i.e.*, the method of accounting] it teaches cannot be used without employing the methods and diagrams used to illustrate the book, or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given to the public."[54]

It is clear from a reading of *Baker*, however, that this passage sets forth a rule of exclusion, not inclusion. It is similar to what later devel-

---

concluded that the "menu command structure" was copyrightable, *see id.* at 68, and had been infringed. *See id.* at 84. It is not clear, however, whether the court was holding the "display screens" (that which appears on the screen), the user interface (what keystrokes, in what order, implement what functions) or both to be protected. At the end of its discussion of copyrightability, the court seems to indicate that both, taken as a whole, constitute expression. *See id.* at 68. Because all expression is copyrightable under conventional copyright doctrine, one would normally assume that either or both are copyrightable under this decision. Due to the court's inexplicable insertion of a substantiality requirement (prong 3), however, it is difficult to determine the court's true holding. *Cf.* Note, *Copyright Protection For Computer Languages: Creative Incentive Or Technological Threat?*, 39 Emory L.J. 1293, 1330-35 (1990)(criticizing *Lotus* on the grounds that it constituted an impermissible extension of copyright protection to computer languages).

50. *See supra* notes 23-25 and accompanying text.

51. *See* Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1234 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

52. 101 U.S. 99 (1879)

53. *Whelan*, 797 F.2d at 1236(emphasis omitted); *see also* Yen, *A First Amendment Perspective on the Idea/Expression Dichotomy and Copyright in a Work's "Total Concept and Feel"*, 38 Emory L.J. 393, 413-14 (1989)(criticizing *Whelan*'s interpretation of a program's idea).

54. *Whelan*, 797 F.2d at 1236 (quoting Baker v. Selden, 101 U.S. 99, 103 (1879)).

oped into the merger doctrine[55] because the Court simply stated that if the accounting form is necessary to practice the system of accounting, it is *not* copyrightable. It does *not* say that everything in Selden's book that is not necessary to the system of accounting is expression. This is the result that would be implied by an acceptance of *Whelan's* interpretation. Under this interpretation, Selden's book would actually receive more copyright protection than a corresponding novel because an infringer must copy significantly more than what is "necessary" to the "purpose" of a novel to infringe on its plot.[56] Nevertheless, *Whelan*, adopting this test, proceeded to declare that the "purpose" of the program at issue was to "aid in the business operations of a dental laboratory."[57] The court therefore concluded that anything not essential to this purpose was protectable expression.[58]

Even if one accepts the court's interpretation of *Baker*, however, its definition of the purpose of the program is dangerously broad. It is akin to saying that the purpose of Selden's book was to create a good accounting system. Taking this proposition to its logical conclusion, Selden's system was protectable expression because there is more than one "good" system of accounting. Clearly, the Court in *Baker* would not agree with this proposition.[59]

III. THE STATUTORY AND PRACTICAL BARS TO SSO PROTECTION

A. *Section 102(b): The Oft-Ignored Bar To SSO Protection*

A significant bar to protection of the SSO of computer programs is Section 102(b) of the Act, which precludes protection of any "procedure, process, system, [or] method of operation."[60] The Judiciary Committee notes accompanying Section 102(b) state:

> Some concern has been expressed lest copyright . . . extend protection to the *methodology or processes adopted by the programmer*, rather

---

55. In fact, in Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738 (9th Cir. 1971), the court used the Supreme Court's decision in *Baker* as precedent for the merger doctrine. *See id.* at 742. For a discussion of merger, see *supra* note 26 and accompanying text.

56. *See* Learned Hand Test, *supra* note 38 and accompanying text.

57. Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1238 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

58. *See id.* at 1238-39.

59. In fact, *Whelan* has been criticized even by supporters of SSO protection. "Perhaps the single virtue of the *Whelan* rule is that it is easy to apply . . . . Given the court's broad conception of the purpose of a program, . . . almost any particular structure could be seen as not necessary to that purpose." Note, *supra* note 13, at 881 (citation omitted); *see also*, Note, *Does Form Follow Function? The Idea/Expression Dichotomy in Copyright Protection of Computer Software*, 35 UCLA L. Rev. 723, 748-49 (1988)(noting that *Whelan* is in conflict with *Baker*); Note, Whelan Associates v. Jaslow Dental Laboratories: *Copyright Protection For Computer Software Structure—What's The Purpose?*, 1987 Wis. L. Rev. 859, 893 (1987)(arguing that *Whelan* must be "refined" by narrowing its interpretation of the "purpose" of a program).

60. 17 U.S.C. § 102(b) (1988).

than merely to the "writing" expressing his ideas. Section 102(b) is
intended . . . to make clear that the expression adopted by the
programmer is the copyrightable element in a computer program, and
that the actual processes or methods embodied in the program are not
within the scope of the copyright law.[61]

This seems to preclude protection of the SSO, which is nothing if not the
method of operation of a computer program.[62] Returning to *Whelan's*
analogy between the SSO and the plot of a book, it is clear that nothing
in Section 102(b) would preclude protection of the plot of a novel. The
Whelan analogy between the SSO of a program and the plot of a novel
therefore fails both legally and conceptually. For these reasons, the pro-
tection of SSO through copyright cannot be defended either through stat-
ute or precedent.[63]

B.   *The Practical Bars To Protection*

1.   Inadvertent "Patents" On The Underlying Hardware

Even if one were to ignore the statutory bar of Section 102(b), policy

---

61. H.R. Rep. No. 1476, 94th Cong. 2d Sess. 57, *reprinted in* 1976 U.S. Code Cong. &
Admin. News 5659, 5670 (emphasis added).

62. This point has not gone unnoticed. As one commentator observed: "it seems un-
deniable that the structure, sequence, and organization of a computer program is pre-
cisely the kind of subject matter . . . that the drafters of Section 102(b) [had] . . . in mind
when they denied protection to methods of operation, procedures, and processes."
Reichman, *Computer Programs As Applied Scientific Know-How: Implications of Copy-
right Protection for Commercialized University Research*, 42 Vand. L. Rev. 639, 696-97
(1989).

Commentators in support of SSO protection inevitably point to another segment of the
same House Report that states: "Section 102(b) in no way enlarges or contracts the scope
of copyright protection under the present law. Its purpose is to restate . . . that the basic
dichotomy between expression and idea remains unchanged." H.R. Rep. No. 1476, 94th
Cong., 2d Sess. 57, *reprinted in* 1976 U.S. Code Cong. & Admin. News 5659, 5670. *See*
Clapes, *Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protec-
tion For Computer Programs*, 34 UCLA L. Rev. 1493, 1548-49, 1553 (1987). This read-
ing of the legislative history, however, ignores the more obvious conclusion that the
legislators recognized that copyright law at that time did not protect SSO and sought
only to codify this aspect of copyright law in order to keep it that way. It should be
remembered that Section 102(b) was in effect for eight years before *any court* recognized
protection of the SSO of computer programs. The court in *Whelan*, while mentioning the
general language of the House Report, completely ignored the language that specifically
related to computer programs. *See* Whelan Assocs., Inc. v. Jaslow Dental Laboratory,
Inc., 797 F.2d 1222, 1234 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

63. In an article published shortly after the *Whelan* decision, one author observed:

   Section 102(b)'s implications for the protection of computer programs are crys-
   tal clear. You need only contrast section 102(b)'s statement of what copyright
   does *not* protect with the same Act's definition of "computer program," to ap-
   preciate how very thin, indeed, is the infringement protection available to com-
   puter software.

Goldstein, *Infringement of Copyright in Computer Programs*, 47 U. Pitt. L. Rev. 1119,
1125 (1986). Goldstein characterized *Whelan* as a decision that "may be supportable on
[its] facts [but] construed the copyright concept of 'idea' too literally . . . . The functions
that *Whelan* was disposed to characterize as protectable expression are more accurately
characterized as unprotectable ideas." *Id.* at 1125-26.

considerations would nevertheless mandate a rejection of SSO protection. A major flaw in many of the arguments in favor of SSO protection is that they fail to recognize that the microprocessor has become a major design element throughout the engineering industry[64] and therefore that the expansion of copyright protection will have ramifications well beyond the computer industry.

The most serious practical problem in protecting the SSO of computer programs is that such protection will inevitably conflict with patent principles with regard to "embedded systems."[65] To understand properly the significance of extending *Whelan*-type protection to such systems, it is necessary first to examine the protection afforded to the hardware in which the program is "embedded." Patent law grants to an inventor of any "process, machine, composition of matter, manufacture, or any improvement thereof" the exclusive rights to the invention for seventeen years, provided that "the claimed invention is new, useful, and unobvious over . . . previously known subject matter."[66]

For an invention to be patentable, therefore, it must meet the tests of novelty and nonobviousness,[67] a much higher standard than that of copyright, which is merely originality and fixation.[68] This represents a basic policy that inventions should not be patented unless they meet a certain qualitative standard.[69] Copyright law, in contrast, imposes no qualitative requirement whatsoever.[70]

The protection of SSO becomes particularly dangerous when applied to embedded systems because it blurs the distinction between patent and copyright law. Several cases involving embedded systems have already been litigated. For example, mobile radios,[71] "fax" machines,[72]

---

64. *See, e.g.,* R. Smith, Electronics: Circuits and Devices 242-43 (3d ed. 1987)(the microprocessor has "completely changed our approach to the design of all kinds of machines used in work and play, at home and in business, in the factory and in the laboratory"); D. McGlynn, Microprocessors, Technology, Architecture, and Applications 22 (1976)(discussing the effect of microprocessors on the electronics industry).

65. For the purposes of this Note, the term "embedded systems" will refer to those devices that are not commonly associated with "computers" but nevertheless contain microprocessors and, therefore, computer programs. Typical examples of these include automobile transmissions, VCRs, TVs, music synthesizers, stereos and industrial equipment. *See* J. Hayes, Digital System Design And Microprocessors 55 (1984).

66. R. Hildreth, Patent Law: A Practitioner's Guide 5 (1988); *see* 35 U.S.C. § 101 (1988).

67. *See* F. Foster & R. Shook, Patents, Copyrights & Trademarks 146 (1989); 35 U.S.C. §§ 102, 103 (1988).

68. *See* 17 U.S.C. § 102 (1988); F. Foster & R. Shook, *supra* note 67, at 144, 146.

69. *See* R. Wincor & I. Mandell, Copyright, Patents And Trademarks: The Protection of Intellectual and Industrial Property 42 (Legal Almanac Series No. 14, 1980).

70. For example, in Alfred Bell & Co. v. Catalda Fine Arts, Inc., Ltd., 191 F.2d 99 (2d Cir. 1951), the court stated: "All that is needed to satisfy both the Constitution and the [copyright] statute is that the 'author' contributed something more than a 'merely trivial' variation, something recognizably 'his own.' . . . No matter how poor artistically the 'author's' addition, it is enough if it be his own." *Id.* at 102-03 (footnotes omitted).

71. *See* E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485 (D. Minn. 1985).

microcode located *within* microprocessor chips that control the internal functioning of the devices,[73] and shot-timer devices used in competitive pistol shooting[74] have all been found to contain copyrightable computer programs.

In embedded systems, the programs are integrally related to the hardware itself.[75] Therefore, while the hardware itself may not meet the requirements for a patent, a copyright on the SSO of the program embedded in the hardware could create pseudo-patent protection of the hardware because that particular hardware configuration must utilize that particular SSO.[76] As a result, the hardware within the embedded

---

72. *See* Secure Servs. Technology, Inc. v. Time and Space Processing, Inc., 722 F. Supp. 1354 (E.D. Va. 1989).

73. *See* NEC Corp. v. Intel Corp., 645 F. Supp. 590 (N.D. Cal. 1986).

74. *See* Pearl Sys., Inc. v. Competition Elecs., Inc., 8 U.S.P.Q.2d (BNA) 1520, Copyright L. Dec. (CCH) ¶26,338 (S.D. Fla. 1988).

75. At this juncture, it is necessary to explore the relationship between hardware and software. In most applications, it is possible for the engineer to choose to implement his design utilizing either hardware, software or a combination of both. Indeed, it has been noted:

> *[H]ardware and software are logically equivalent.*
>
> Any operation performed by software can also be built directly into the hardware, and any instruction executed by the hardware can also be simulated in software. The decision to put certain functions in the hardware and others in the software is [based on] such factors as cost, speed, . . . reliability, and frequency of expected changes.

A. Tanenbaum, Structured Computer Organization 10 (1976).

Hardware and software are, therefore, interchangeable elements in an engineer's arsenal. In most "embedded" systems, they are interdependent elements of the overall device. *See, e.g.,* McGlynn, *supra* note 64, at 32 (When developing a prototype system the "interplay between hardware and software . . . must be carefully synchronized."); W. Clark, The Microprocessor and its Application 90 (1978)(the solutions to most design problems require a combination of both hardware and software).

76. In discussing NEC Corp. v. Intel Corp., 645 F. Supp. 590 (N.D. Cal. 1986), one author noted that experts for both parties "testified that if the underlying hardware were the same, the range of expression available to [the programmer] would have been limited substantially." A. Clapes, Software, Copyright and Competition 127 (1989). *Accord* NEC Corp. v. Intel Corp., 10 U.S.P.Q.2d 1177, 1188-89, Copyright L. Dec. (CCH) ¶ 26,379, at 22,389 (N.D. Cal. 1989). Thus, experts for both parties admitted that the hardware and software elements of the device were inextricably intertwined. Clapes, however, appears to have overlooked the implications of these statements with regard to embedded systems.

For example, E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485 (D.C. Minn. 1985), involved a mobile radio that included in its design a microprocessor and its accompanying software. *Id.* at 1488. The radio as a whole presumably did not meet the qualitative standard necessary for a patent. The software, however, was entitled to copyright protection.

In this case, the court found infringement because the defendant had "translat[ed] the EFJ code virtually verbatim from Intel [the plaintiff's microprocessor] to Hitachi [the defendant's microprocessor] code." *Id.* at 1496. This is consistent with copyright protection of the literal elements of a program and is a just result because the defendant could have implemented the processes embodied in the program without copying the plaintiff's code.   If a similar case were to be litigated after *Whelan,* however, the plaintiff would be entitled to protection not only against copying of the literal elements of the program, but also against copying of the SSO of the program. Due to the interdependence between

system would be protected from copying for the duration of the copyright.[77] Thus, the copyright protection of SSO could allow inventors to frustrate one of the aims of patent law, which is to "promote the Progress of Science,"[78] by allowing such copying.[79]

2. Lack of Disclosure and Notice Requirements

The changes in the Act with respect to registration and notice provide further reason for concern. Under the Act, copyright protection "subsists . . . in original works of authorship," meaning that protection begins at the moment of creation, provided that the work is "fixed" in a "tangible medium of expression."[80] Therefore, the author need not register a work with the Copyright Office to receive protection. It is necessary only for an author to register a work before initiating an infringement action.[81] Moreover, beginning March 1, 1989, an author was no longer required to put a notice of copyright on his work.[82]

---

hardware and software in such systems, it would probably be impossible for the defendant to use the plaintiff's unprotected hardware without infringing on the plaintiff's protected SSO. The doctrine of merger might not act to bar this result because the defendant need not use either that hardware configuration or plaintiff's SSO to achieve his goal of creating a mobile radio that, as a whole, functions in an identical manner as plaintiff's radio. Therefore, plaintiff's SSO is arguably not the only way to express plaintiff's "idea"—a radio with plaintiff's specifications—and there is no merger. For a discussion of merger, see *supra* note 26 and accompanying text.

77. The duration of copyright protection is another significant factor to be considered. Patent law provides the inventor with the right to exclude others from making, using and selling the invention for seventeen years from the issuance of the patent. *See* 35 U.S.C. § 154 (1988). Copyright law, on the other hand, provides the author with the exclusive right to produce and distribute the work for the life of the author plus fifty years. *See* 17 U.S.C. §§ 106, 302, 303 (1988). Therefore, copyright protection of the SSO of computer programs may not only inadvertently extend protection to accompanying hardware but may also provide protection that is significantly longer in duration than patent protection.

78. U.S. Const. art. I, § 8, cl. 8.

79. The patent law requirement that, in order to be patentable, an article be new, useful and nonobvious, *see* 35 U.S.C. §§ 101, 103 (1988), necessarily provides that inventions that are "obvious . . . to a person having ordinary skill in the art," 35 U.S.C. § 103 (1988), receive no patent protection and are dedicated to the public, and thus, available for copying.

Consequentially, it is common industry practice to dissect a competitor's product in order to copy its desired aspects. Examples of this practice have been referred to in several cases. *See, e.g., E.F. Johnson*, 623 F. Supp. at 1489-90 ("Uniden commenced development of a mobile radio compatible with EFJ's LTR radios and repeaters sometime in 1984. In the course of developing an LTR-compatible radio, Uniden . . . disassembled the software found in EFJ's model 8855 and 8800 radios."); NEC Corp. v. Intel Corp., 645 F. Supp. 590, 592 (N.D. Cal. 1986)(noting similarities in the hardware configurations).

80. 17 U.S.C. § 102(a) (1988).

81. *See* 17 U.S.C. § 411 (1988). A work, however, need not be registered at the time of the infringement: "a copyright owner who has not registered his [work] can have a valid cause of action against someone who has infringed his copyright, but he cannot enforce his rights in the courts until he has made registration." H.R. Rep. No. 1476, 94th Cong., 2d Sess. 157, *reprinted in* 1976 U.S. Code Cong. & Admin. News 5659, 5773.

82. *See* 17 U.S.C. § 401 (1988).

The provisions with respect to registration and notice enable the creator of a program to receive copyright protection for his work *without* disclosing the work to the public.[83] In the case of an "embedded system," the copyright protection of SSO will therefore result not only in the de facto protection of previously unprotectable hardware, but also in protection of the system as a whole without complying with the disclosure requirements of patent law.[84]

In addition, the removal of the notice requirement allows an author to receive protection without putting the public on notice that the device at issue is claiming copyright protection. This is not a problem for most copyrightable works because it is assumed, for example, that a novel is subject to protection. As technology continues to blur the line between hardware and software, however, it is becoming increasingly difficult to tell what *is* copyrightable.[85] A competitor is thus left in the unenviable

---

83. Copyright law "reflects a balance of competing claims upon the public interest: Creative work is to be encouraged and rewarded, but private motivation must ultimately serve the cause of promoting broad public availability of literature, music, and the other arts." Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156 (1975). In all types of protected works except computer programs, distribution of the work creates disclosure of its contents. In the area of computer programming, however, distribution is accomplished without disclosure. Congress, by allowing computer programs to be considered literary works, presumably determined that the need to protect software outweighed the need for full public disclosure. As expansion of the scope of protection is considered by the courts, however, this balance must be reconsidered. *See, e.g.,* Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form,* 1984 Duke L.J. 663, 705-19 (arguing that disclosure is an important policy of copyright law and asserting that copyright protection of computer programs subverts this policy); Davidson, *Common Law, Uncommon Software,* 47 U. Pitt. L. Rev. 1037, 1056-57 (1986)(admitting that lack of public disclosure arguably subverts public policy, but claiming that the Act is concerned with authorship, not disclosure). *But see* Raskind, *supra* note 7, at 1140-41 (asserting that copyright law is not concerned with disclosure).

84. As support for this proposition, consider a system built by Corporation A in which the hardware and software are integrally related. *See supra* note 75. If the device is patented, Corporation A must fully disclose the device in the patent. *See* 35 U.S.C. § 111, 112 (1988). In return for this disclosure, Corporation A's device will be protected against copying. In the absence of patent and SSO protection, however, Corporation B is permitted to copy the device, provided that it does not copy the literal elements of the device's software. If SSO protection is extended to this device's software, however, Corporation A will be able to prevent copying of both the hardware and software elements of its device without *any* disclosure. *See supra* note 76. Thus, with respect to an unpatentable device, Corporation A can receive what is otherwise unattainable with a patentable device—protection of the entire device without disclosure.

85. An excellent illustration of the "blurring" of the line between hardware and software is the ASIC, *see supra* note 6, which one commentator has asserted is copyrightable. *See* Lunney, *supra* note 6. Lunney notes that "[t]o create an ASIC, the designer writes a set of descriptive statements in either Boolean algebra or a simulator's descriptive language." *Id.* at 176. He therefore claims that these statements are within the Act's definition of a computer program because "[t]hey are a set of statements written to be used in a computer . . . to produce a result, a final gate configuration." *Id.* at 177. Thus, by defining an ASIC in terms of the "program" that creates it, copyright protection could arguably extend to custom and semi-custom logic devices. An engineer in the industry, however, would never imagine that such a device could be subject to copyright protection

714 FORDHAM LAW REVIEW [Vol. 59

position of having to guess what portions of a device are protected.

IV. COPYRIGHT PROTECTION OF A PROGRAM'S LITERAL ELEMENTS COMBINED WITH PATENT PROTECTION OF A PROGRAM'S STRUCTURE IS ADEQUATE

The pitfalls of protecting SSO through copyright can be avoided by allowing patent law to protect any program structure that meets the patent requirements of novelty and nonobviousness.[86] Copyright protection, which extends to all programs, should be limited to the program's literal elements. All programmers, therefore, would receive protection against literal copying, while only those programers whose work meets the qualitative standards of patent law would receive protection of their work's structure. The remainder of this Note will be devoted to exploring, first, the scope of copyright protection for the program's literal elements, and second, the scope of patent protection for the program's structure.

In order to comprehend the *scope* of literal copyright protection it is necessary to understand the distinction between copyrightability and infringement. All expression is copyrightable. Copyright protection therefore subsists even in a small portion of code. In order to sustain an infringement action, however, one must prove "ownership of a valid copyright and unauthorized copying of protected material . . . *that goes so far as to constitute unlawful appropriation.*"[87] The literal copying of a small portion of a program can therefore support a copyright infringement action if the amount of code copied constitutes the heart of the work.[88]

In addition, the literal protection of a program includes the literal translation of a program from one computer language to another.[89] This level of protection, combined with the protection of patent law, is adequate to serve the goal of intellectual property law, which is to "promote

---

and would assume that, because the device had no patent notice, he was free to copy it. *See also* Mislow, *Computer Microcode: Testing the Limits of Software Copyrightability,* 65 B.U.L. Rev. 733, 740 (1986)(discussing the copyrightability of computer microcode).

86. *See supra* notes 66-67 and accompanying text.

87. W. Patry, Latman's The Copyright Law 191 (6th ed. 1986)(emphasis added). Three distinct elements must be shown to succeed in an infringement action. The first, ownership of a valid copyright, has been the focus of this Note. The second element, copying, can be shown either directly (i.e., plaintiff produces a witness who saw the defendant copy the work) or circumstantially. *See id.* In order to show circumstantial copying, the plaintiff must prove access plus substantial similarity between the copyrighted work and the infringing work. *See id.* at 192-96. The third requirement is that the material copied be enough to constitute an unlawful appropriation. This element can be satisfied through copying of a quantitatively substantial part of the work, or through copying of a qualitatively significant part of the work—the heart of the work. *See id.* at 196 & n. 31. A plaintiff, therefore, could succeed in an infringement action by proving the literal copying of a small part of a program if the copied portion was substantial in a qualitative sense.

88. *See supra* note 87.

89. *See* E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1497-98 (D. Minn. 1985).

the Progress of Science and useful Arts."[90]

Although the scope of patent protection of computer software is inherently limited by the nature of patent law itself, patent protection is available to the SSO of computer programs. Due to the qualitative standard any invention must meet to be patentable, however, most computer programs will not be afforded protection. This hurdle is not, however, impossible to overcome.[91]

In *Paine, Webber, Jackson & Curtis, Inc. v. Merrill Lynch, Pierce, Fenner & Smith, Inc.,*[92] the court upheld the patentability of a program relating to a data processing methodology for cash management accounts.[93] Similarly, in *In re Toma,*[94] the Court of Customs and Patent Appeals held that a program that "translate[d] from a source natural language, e.g., Russian, to a target natural language, e.g., English" was patentable.[95]

Programs that essentially perform mathematical algorithms are subject to additional scrutiny, however, because formulas in general are not patentable.[96] The general rule is that "[i]f the claimed invention is a mathematical algorithm, it is improper subject matter for patent protection, whereas if the claimed invention is an application of the algorithm [it may be patentable subject matter]."[97]

The patentability of computer programs, like their copyrightability, is

---

90. U.S. Const. art. I, § 8, cl. 8. In fact, at least one supporter of SSO protection has implicitly endorsed this conclusion:

> To deny all protection of structure would theoretically reduce the incentives to create new, innovative structures. This reduction would, however, be small, because the desire for efficient implementations of program functions is a powerful incentive to create new structures.

Note, *supra* note 13, at 896.

91. *See, e.g.,* Smith, *Patenting Computer-Related Inventions,* 1 Software L.J. 33, 34, 38-44 (1985)(discussing various types of computer programs and instructing the reader as to the best method for patenting them).

92. 564 F. Supp. 1358 (D. Del. 1983).

93. *See id.* at 1369. The court noted that computer programs are recognized as being patentable and must "meet the same requirements as other inventions in order to qualify for patent protection." *Id.* at 1366.

94. 575 F.2d 872 (C.C.P.A. 1978).

95. *Id.* at 874.

96. The Supreme Court, in a case involving a data processing method for converting binary-coded decimal numbers into standard binary numbers, stated:

> It is conceded that one may not patent an idea. But in practical effect that would be the result if the formula for converting BCD numerals to pure binary numerals were patented in this case. The mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself.

Gottschalk v. Benson, 409 U.S. 63, 71-72 (1972). *See also* Diamond v. Diehr, 450 U.S. 175, 185-93 (1980)(interpreting *Benson*); Smith, *supra* note 91 (discussing the patenting of computer software).

97. Smith, *supra* note 91, at 38 (emphasis omitted)(construing *In re Abele,* 684 F.2d 902, 907 (C.C.P.A. 1982)).

a controversial issue.[98] Even opponents of patent protection, however, admit that computer programs are currently patentable.[99] Therefore, to avoid conflicts with patent law, the SSO of computer programs should be protected solely through patents, while the literal elements of computer programs should continue to be protected through copyright law. This solution will provide adequate protection for computer programs without sacrificing the interests of patent law with regard to the underlying hardware.

CONCLUSION

The recent trend toward extension of copyright protection to the SSO of computer programs is flawed on several grounds. First, it rests on an erroneous interpretation of *Baker v. Selden*. Second, even if this interpretation is accepted, there is a clear statutory bar under Section 102(b) of the Act. Third, even in the absence of this bar, SSO protection must be rejected on policy grounds due to its inevitable encroachment on patent law with respect to "embedded" systems. The proper course is to continue to provide copyright protection to the literal elements of computer

---

98. It has been asserted by at least one commentator that patent law protects only "software linked to a physical process." Note, *Computer Intellectual Property and Conceptual Severance*, 103 Harv. L. Rev. 1046, 1051 (1990)(construing Diamond v. Diehr, 450 U.S. 175 (1980)). This assertion, however, is misleading. The holding of *Diehr* applies only to programs that recite *mathematical algorithms*. *See Diehr*, 450 U.S. at 192-93. The current approach to these types of programs is:

> The method adopted by this court for analyzing mathematical algorithm-statutory subject matter cases comprises a two part test: First, the claim is analyzed to determine whether a mathematical algorithm is directly or indirectly recited. Next, *if a mathematical algorithm is found*, the claim as a whole is further analyzed to determine whether the algorithm is "applied in any manner to physical elements or process steps," and, if it is, it "passes muster under [section] 101."

In re Pardo, 684 F.2d 912, 915 (C.C.P.A. 1982)(emphasis added)(citations omitted). Thus, it appears that *Diehr* has had no effect upon the patentability of programs that do not recite mathematical algorithms. If it were true that all programs must be applied to a physical process, there would be no need for the first prong of the *Pardo* test.

99. In fact, one author who opposes patent protection for computer programs admits that "[a]fter *Diehr*, the Patent Office ceased to resist issuing patents for computer program-related inventions. It now only rarely rejects applications pertaining to those inventions on subject matter grounds." Samuelson, Benson *Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 Emory L. Rev. 1025, 1093-94 (1990). While Ms. Samuelson disagrees with the *Toma* and *Merrill Lynch* decisions, *see supra* notes 93-94 and accompanying text, she admits that:

> Although these opinions recognize that claims for . . . "mathematical algorithms" are unpatentable under *Benson*, they hold that claims for other kinds of information processing algorithms are patentable *because they can be carried out by machine* and have not been made unpatentable by *Benson*.

Id. at 1122-23 (emphasis added); *see, e.g.,* Chisum, *The Patentability of Algorithms*, 47 U. Pitt. L. Rev. 959, 1019-20 (1986)(supporting patent protection even for programs that implement mathematical algorithms); Menell, *An Analysis of the Scope of Copyright Protection For Application Programs*, 41 Stan. L. Rev. 1045, 1075 (1989)("[t]he Supreme Court's 1981 decisions . . . cleared the way for patent protection for computer programs"); Smith, *supra* note 91, at 33 (a detailed practical analysis of the patentability of computer programs).

programs and allow patent law to extend protection to computer program structure.

Cary S. Kappel

.